Science Project Detailed Research Plan

Please complete the information/questions in red ink below. Save your plan to your computer and upload a copy to your application Forms Folder.

Date: Jan 8, 2025
Student Name: Pranav Sethuraman, Derek Yang, Abhinav Namboori

Project Title: :Evaluating the effect of Physics-Informed Neural Networks (PINNs) on the predictive accuracy and real-time conjunction risk assessment of Low-Earth Orbit (LEO) debris.

My project Title is: Physics-Informed Neural Networks for High-Fidelity Orbital Propagation and Real-Time LEO Conjunction Assessment

Low Earth Orbit has become increasingly congested due to the rapid growth of commercial satellite constellations, legacy spacecraft, and fragmented debris from past collisions and rocket launches. This problem significantly raises the risk of orbital conjunctions, meaning close approaches or collisions between objects, which can generate cascading debris events known as Kessler Syndrome and potentially render parts of Earth orbit unusable for decades. Current operational tracking systems rely heavily on analytical propagation models such as SGP4, which is computationally efficient, but suffer from increasing positional uncertainty over time due to atmospheric drag, solar activity, and Earth's non uniform gravitational field. As a result, satellite operators must perform thousands of costly and unnecessary collision avoidance maneuvers each year based on conservative risk estimates.

Hypothesis: Based on your reading and information research, organize everything you have discovered, and then make an estimate of what will happen. Knowing certain things are true, you then predict what might happen if you change something. Your experiment, when successful, will allow you to determine if your hypothesis was correct or not.

My project Hypothesis is: If a Physics-Informed Neural Network (PINN) is used for orbital propagation, where the loss function is constrained by the partial differential equations of orbital dynamics, then the model will demonstrate a statistically significant reduction in Euclidean position error compared to the SGP4 baseline.

Materials: List all necessary biological agents, chemicals, reagents, major instruments, and software which will be used.

My project Materials are listed below:

Hardware & Major Instruments:

- High-Performance Computing (HPC) Workstations: Three laptop/desktop computers (Intel i7/AMD Ryzen 7, 16GB RAM).
- Graphics Processing Unit (GPU): NVIDIA RTX series GPU for AI training.
- High-Speed Internet Access: For REST API data downloading.

Software & Programming Environments:

- Operating Systems: Windows 11 / macOS.
- Development Environment: Visual Studio Code and GitHub for version control.
- Programming Language: Python 3.10+.

Libraries & Data Sources:

- AI Frameworks: PyTorch and NumPy (for PINN construction and matrix math).
- Astrodynamics Libraries: Astropy, Skyfield, and the SGP4 Python library (Control Group).
- Data Sources: Space-Track.org (US Space Command Catalog) and CelesTrak (Space Weather/Solar Flux data).

Methods: Describe the **general methods** to be used, and why are you using the methods you have chosen? Why have you chosen the described controls? Examples would **be spectroscopy, photometric methods, direct measurement, volume displacement, voltage, energy output, etc. WHAT IS YOUR MEASURED END POINT(S)?**

My general project methods are:

Our project aims to enhance space junk tracking through computational physics modeling. We selected a technique known as Physics-Informed Neural Networks (PINNs) because conventional AI frequently makes "dumb" predictions that defy the laws of physics, such as forecasting that a satellite will abruptly take off into deep space. By directly programming real orbital formulas into the AI's "brain," we compel it to adhere to scientific principles as it gains knowledge from actual NASA data. Since the SGP4 model is currently NASA's standard math, we are using it as our control. If our AI can predict a satellite's path more accurately than SGP4, we will demonstrate the superiority of our approach. We also use probability bubbles to represent the "risk zone" around a piece of debris, and a 3D sorting trick called an Octree to make our code run in real-time.

The Euclidean Distance Error, which is simply the distance in kilometers between where our AI predicts a satellite will be and where it actually ends up, is our main measured end point. The risk of Kessler Syndrome, a series of collisions that could make space unusable for generations, is at an all-time high as commercial satellite "mega-constellations" pack the Low-Earth Orbit (LEO). Due to erratic atmospheric drag, current tracking models frequently have "blind spots," requiring satellite operators to carry out thousands of needless and expensive "collision avoidance maneuvers" annually. In order to ensure the long-term viability of space exploration, our research offers a higher-precision tool that enables operators to differentiate between a "false alarm" and a deadly threat.

Detailed Experimental Procedure: State your DETAILED methods, so that others could repeat your work **exactly**. Include details, giving exact specifications and quantities. [Your procedure will describe how you plan to do your experiment, changing only one variable at a time and keeping all the other parameters the same]. Describe your control so that you can compare results of your experiment with a standard for which the variable is unchanged. Make sure that you have three or more seeds/plants/animals in each of the control and experimental groups. Even better, have several experimental groups (e.g. more than one concentration of chemical you are testing, more than one time point, etc). Make measurements in metric units when possible. Repeat the test more than once to see if your results are reproducible.

My DETAILED project methods are:

1. Data Preparation and Grouping: We will first gather a massive dataset of satellite positions from Space-Track.org. Our "test subjects" consist of 35,000 distinct objects in Low-Earth Orbit (LEO), including 10,500 active satellites, 4,500 rocket bodies, and 20,000 fragments of debris. We will divide our experiment into three groups to test the impact of our variable:

- Control Group: Predictions made using the industry-standard SGP4 model (no AI involved).
- Experimental Group A (Basic AI): Predictions made using a standard Neural Network (no physics laws).
- Experimental Group B (PINN): Predictions made using our Physics-Informed Neural Network (AI + orbital mechanics).

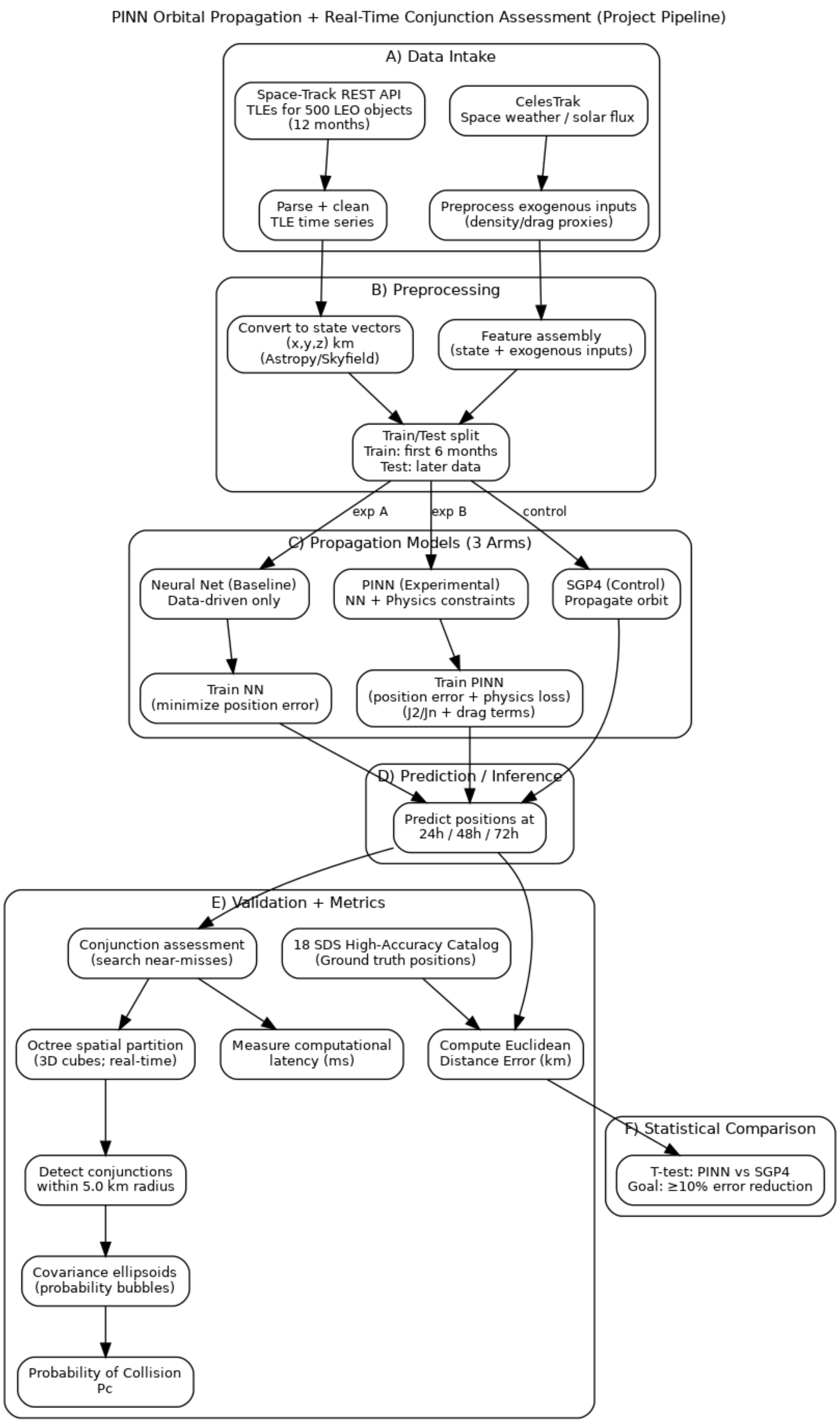
2. Step-by-Step Execution:

1. Scripting: Write a Python script to download Two-Line Element (TLE) data for the 500 objects over a 12-month period.
2. Coordinate Conversion: Use the Astropy library to convert raw TLE data into metric state vectors (x,y,z) in kilometers.
3. Training: The Coder will train the AI models using the first 6 months of data. The Mathematician will code the "Loss Function" for Group B, penalizing the AI if it violates the J2 gravity perturbation formulas.
4. Propagation: We will run all three groups to predict where each of the 500 objects will be at exactly 24, 48, and 72-hour intervals in the future.
5. Collision Search: We will apply an Octree spatial algorithm to partition space into 3D cubes. We will measure how many near-misses within a 5.0 km radius each model detects.

3. Measurement and Repeating:

- Metric Measurement: For every prediction, we will calculate the Euclidean Distance Error in kilometers by comparing the model's guess to the actual recorded position from the Space-Track catalog.
- Repetition: To ensure our results are reproducible and not just "lucky guesses," we will repeat this entire test cycle across four different weeks in the year to account for varying solar activity and atmospheric density changes.
- Uncertainty Calculation: We will generate Covariance Ellipsoids (3D probability bubbles) for each prediction to visualize the "risk zone" where a crash could occur.

4. Statistical Analysis: We will use a T-test to compare the average error of our PINN (Group B) against the Control Group (SGP4). Our goal is to see if the PINN reduces the error by at least 10% consistently across all 35,000 objects.


PINN Orbital Propagation + Real-Time Conjunction Assessment (Project Pipeline)
A) Data Intake
Space-Track REST API TLEs for 500 LEO objects (12 months)
CelesTrak Space weather / solar flux
Parse + clean TLE time series
Preprocess exogenous inputs (density/drag proxies)
B) Preprocessing
Convert to state vectors (x,y,z) km (Astropy/Skyfield)
Feature assembly (state + exogenous inputs)
Train/Test split Train: first 6 months Test: later data
exp A
exp B
control
C) Propagation Models (3 Arms)
Neural Net (Baseline) Data-driven only
PINN (Experimental) NN + Physics constraints
SGP4 (Control) Propagate orbit
Train NN (minimize position error)
Train PINN (position error + physics loss) (J2/Jn + drag terms)
D) Prediction / Inference
Predict positions at 24h / 48h / 72h
E) Validation + Metrics
Conjunction assessment (search near-misses)
18 SDS High-Accuracy Catalog (Ground truth positions)
Octree spatial partition (3D cubes; real-time)
Measure computational latency (ms)
Compute Euclidean Distance Error (km)
Detect conjunctions within 5.0 km radius
Covariance ellipsoids (probability bubbles)
Probability of Collision Pc
F) Statistical Comparison
T-test: PINN vs SGP4 Goal: ≥10% error reduction

Methods of Data Collection: If you used a published method, reference the method, but describe any changes you made to it. If you used experimental organisms, identify them by genus and species. If you used a standard instrument, it suffices merely to name it, but if you devised a new or special method, describe it completely.

My project uses the following method of data generation:

Our project utilizes the Two-Line Element (TLE) set format as the primary data source, which is the international standard for describing the orbital states of Earth-orbiting objects, collected via the Space-Track.org REST API. We use the SGP4 propagation method as our baseline control, but we introduce a novel computational deviation: a Physics-Informed Neural Network. Unlike traditional data-driven models that only analyze coordinates, our "special method" modifies the AI's learning process by adding a Custom Physics Loss Function based on the Gaussian Perturbation Equations. This allows the model to account for Earth's non-spherical gravity (specifically J2 through J5 zonal harmonics) and atmospheric drag using the Harris-Priester density model. To process this data, we employ several standard computational "instruments," including Astropy and Skyfield for converting TLEs into Cartesian state vectors in the GCRF (Geocentric Celestial Reference Frame), and PyTorch for constructing the neural network. To achieve real-time performance, we devised a recursive 3D Octree spatial partitioning method that allows the system to ignore distant debris and focus solely on objects within the same spatial cube. Finally, all AI predictions are validated against the 18th Space Defense Squadron (18 SDS) High-Accuracy Catalog, which serves as our "ground truth" for measuring model error in kilometers.

Bibliography: List the authors and titles of five, (high school) or three (middle school) science or engineering books or articles that you have read and found useful for your research subject.
Example: Author 's Name, Year of publication, "Quoted Title of Magazine Article (magazines only)"; Underlined *Title of Book or Magazine, date, volume, and number of magazine issue. Page numbers read. If you use a web site: www.urlname.ext, name of topic from the home page, author, and date read.*

My bibliographic references are the following:

1. **Raissi, Maziar; Perdikaris, Paris; and Karniadakis, George E.**, 2019, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations"; *Journal of Computational Physics*, Vol. 378, pp. 686-707. *Accessed Dec 28, 2025.*
2. **Vallado, David A.**, 2013, *Fundamentals of Astrodynamics and Applications*, Fourth Edition, Microcosm Press. Pages 50-150. *Accessed December 28, 2025.*
3. **Kessler, Donald J. and Cour-Palais, Burton G.**, 1978, "Collision frequency of artificial satellites: The creation of a debris belt"; *Journal of Geophysical Research*, Vol. 83, No. A6. pp. 2637-2646. *Accessed Jan 3, 2026.*
4. **Caldas, Filipe M. and Soares, Carlos**, 2022, "Machine Learning in Orbit Estimation: A Survey"; *Acta Astronautica*, Vol. 191, pp. 481-502. *Accessed Jan 3, 2026.*
5. **Hawking, Stephen**, 1988, *A Brief History of Time*, Bantam Books. pp. 15-40. *Accessed Jan 3, 2026.*
6. **Goodfellow, Ian; Bengio, Yoshua; and Courville, Aaron**, 2016, *Deep Learning*, MIT Press. pp. 164-220. Accessed Jan 4, 2026.
7. **Montenbruck, Oliver and Gill, Eberhard**, 2000, *Satellite Orbits: Models, Methods and Applications*, Springer. pp. 55-100. Accessed Jan 5, 2026.

8. **Greene, Brian**, 2004, *The Fabric of the Cosmos: Space, Time, and the Texture of Reality*, Alfred A. Knopf. pp. 47-75. Accessed Jan 5, 2026.